递归、枚举、模拟、递推

2018年12月12日 19:15

- 枚举、模拟、递推

一. 枚举

枚举形式	状态空间规模	一般遍历方式
多项式	n^k，k常数	循环、递推
指数	k^n，k常数	递归、位运算
排列	n!	递归、next_permutation
组合	nCm	递归+剪枝

- 递归

二. 递归的宏观描述

1. 不断缩小问题状态空间的规模，，直至找到问题边界，再回溯，还原现场
2. eg：递归求1~n的集合的幂集
   i. 指数级枚举，O（2^n）

```
vector<int>chosen;
void calc(int x){
    if(x==n+1){//问题边界
        for_(i,0,chosen.size())//换成n会运行时错误，迷
            printf("%d ",chosen[i]);
        puts("");//换行
        return;}
    calc(x+1);//不选x
    chosen.push_back(x);//选x
    calc(x+1);
    chosen.pop_back();}//回溯后还原现场
int main(){calc(1);}//入口
```

3. eg：递归剪枝求元素数m的子集
   i. 组合级枚举，O（nCm）
   ii.

```
vector<int>chosen;
void calc(int x){
    if(chosen.size()>m || chosen.size()+n-x+1<m)
        return;//数量超了，或全选也不能凑够m了，需要剪枝
    if(x==m+1){//问题边界
        for_(i,0,chosen.size())
            printf("%d ",chosen[i]);
        puts("");//换行
        return;}
    calc(x+1);//不选x
    chosen.push_back(x);//选x
    calc(x+1);
    chosen.pop_back();}//回溯后还原现场
int main(){calc(1);}//入口
```

4. eg：递归输出全排列

i. 排列级枚举，O（n！）

ii.
```
int order[20];//输出顺序
bool chosen[20];//选择了与否
void calc(int k){
    if(k==n+1){
        for_(i,0,n)
            printf("%d ",order[i]);
        puts("");
        return;}
    for__(i,1,n){
        if(!chosen[i]){
            order[k]=i;
            chosen[i]=true;
            calc(k+1);
            chosen[i]=false;}}}//还原现场
int main(){calc(1);}
```

三. 递归的机器实现

1. 将参数压入栈，调用汇编call(address)指令将下条语句压入栈，函数返回后调用ret指令将返回地址弹出栈